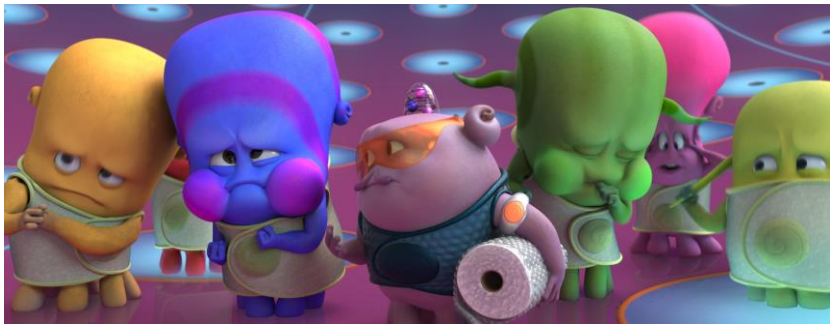
Emoting Boov in *HOME*

Amaury Aubel
DreamWorks Animation



In the animated movie *HOME*, the Boov, an alien species, have a unique way of displaying their emotions. They transition to a specific color depending on their mood: red when angry, yellow when scared, blue when sad, etc. This talk focuses on the visual development of this "emoting effect" as well as how we produced close to 400 shots.

Visual Development

The creative leadership heavily referenced the cuttlefish. This marine animal has the amazing ability to quickly change color and patterns for camouflage. We decided that the emoting effect would consist of a color change and/or moving patterns on the character's skin. The other creative direction was driven by the shape language of *HOME*: the Boov are associated with round, circular, graceful shapes, while their enemies, the Gorg, are characterized by sharp-edged, pointy, triangular visuals.

Early animation tests combined color changes with circular skin patterns that mimicked the characteristic soap bubbles used by the Boov. We realized that keeping the center of the face mostly clear of patterns helped the overall readability and that patterns were most effective when they augmented the facial expressions.

Because the default character's UV set is broken into multiple patches and thus not practical for animation purposes, we created a small library of ancillary "flow UV domains" over which colors and patterns could flow seamlessly (some of these UV sets still had seams but they were strategically hidden under clothing). These "flow" UVs gave us a way to control how emotion colors would travel as opposed to just fading in or out. They also provided blending masks to create "half-emotions" where both the neutral and the emoting color would coexist in the final composite.

The seven main emotion colors were defined by ensuring that their saturation and brightness levels would stay comparable to those of the neutral skin color. Tests of slow color shifts confirmed expected issues in transition areas: these could look muddy or grayish. We thought about using predefined circular (rather than linear) paths in the color wheel but settled on a more general solution: we reworked the character's shading network to accept as input any RGB values for the diffuse and subsurface components, thus allowing the animator to generate any intermediate transition color as needed.

We tried simulating patterns that would travel subcutaneously. R&D developed a parallax shader, inspired by [OAT06], that simulates depth by filtering texels and offsetting texture coordinates based on the intersection of the viewing ray and a virtual surface below the actual one. The shader suffers from discontinuities at seams but our "flow" UVs address that problem. While we produced interesting results, the patterns started to look too much like veins and capillaries and we abandoned that idea.

The main cue we used to drive the emoting effect was the character animation. For patterns, additional elements were useful: for example the direction in which the character was moving or using the patterns to focus the viewer's attention.

We found that crowds turning yellow or red uniformly looked a bit unnatural. Our "flow" UV sets were used to create variations where the original skin color would show up in some areas. This broke the uniformity and made the crowds sit better.

Animation Pipeline

Our animation pipeline was designed to handle the high volume of shots. We started animating patterns and color changes in Houdini but waiting for 3D renders proved unpractical and we quickly migrated as much as possible to Nuke for speed. We set up additional "emoting AOVs" including flow UV mattes, body parts mattes (head, arms, legs) and emotion beauty renders. An install script ensured that these AOVs would get generated automatically and that emoting nodes would be added to the lighter's Nuke script. Using a combination of animation variables and curves, the FX artist controlled when the effect would be activated in lighting and which AOV's would be generated by a lighting render. This proved a great time-saver as FX artists could quickly iterate on light slap comps. Ultimately, most color flushes and a good number of patterns could be generated directly in Nuke. Reflections were also handled in Nuke by setting up equivalent reflections of emoting AOVs. Finally, crowds were animated in a simpler way using curves that would directly drive the lighting renders.

References

OAT, C., 2006. Advanced Real-Time Rendering in 3D Graphics and Games. ACM SIGGRAPH 2006 Course #26 Notes, August.